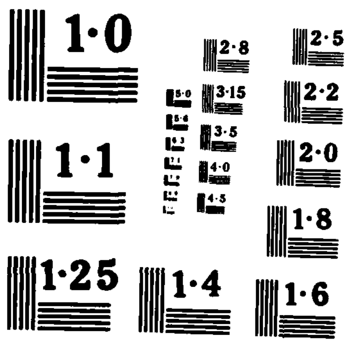
AD-A158 728 APPROACHES TO CONCEPTUAL CLUSTERING(U) CALIFORNIA UNIV IRVINE DEPT OF INFORMATION AND COMPUTER SCIENCE D FISHER ET AL. 12 JUL 85 UCI-ICS-85-17 1/1

UNCLASSIFIED N00014-84-K-0345 F/G 9/4 NL

END
FILMED
DTIC



NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART

2

Information and Computer Science

APPROACHES TO CONCEPTUAL CLUSTERING

Douglas Fisher and *Pat Langley*

Irvine Computational Intelligence Project
Department of Information and Computer Science
University of California, Irvine, California 92717

TECHNICAL REPORT

UNIVERSITY OF CALIFORNIA
IRVINE

APPROACHES TO CONCEPTUAL CLUSTERING

Douglas Fisher and *Pat Langley*

Irvine Computational Intelligence Project
Department of Information and Computer Science
University of California, Irvine, California 92717

Technical Report 85–17

July 12, 1985

We would like to thank Dan Easterlin, whose terminology we partially adopted in describing conceptual clustering processes, and Dennis Kibler, who pointed us to Vere's Thoth system as a possible basis for future work.

To appear in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, 1985.

This research was supported by Contract N00014–84–K–0345 from the Information Sciences Division, Office of Naval Research. Approved for public release; distribution unlimited. Reproduction in whole or part is permitted for any purpose of the United States Government.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report No. 1	2. GOVT ACCESSION NO. AD-A158 728	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Approaches to Conceptual Clustering		5. TYPE OF REPORT & PERIOD COVERED Interim Report 1/85 - 6/85
		6. PERFORMING ORG. REPORT NUMBER 85-17
7. AUTHOR(s) Douglas Fisher and Pat Langley		8. CONTRACT OR GRANT NUMBER(s) N00014-84-K-0345
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Information and Computer Science University of California Irvine, California 92717		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Information Sciences Division Office of Naval Research Arlington, Virginia 22217		12. REPORT DATE 12 July, 1985
		13. NUMBER OF PAGES 14
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

To appear in Proceedings of the Ninth International Joint Conference on Artificial Intelligence, 1985.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

conceptual clustering	learning from examples
concept formation	heuristic search
numerical taxonomy	hierarchical classification

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

(over)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20.

Abstract

Methods for *Conceptual Clustering* may be explicated in two lights. Conceptual Clustering methods may be viewed as extensions to techniques of *numerical taxonomy*, a collection of methods developed by social and natural scientists for creating classification schemes over object sets. Alternatively, conceptual clustering may be viewed as a form of *learning by observation* or *concept formation*, as opposed to methods of *learning from examples* or *concept identification*. In this paper ~~we~~ survey and compare a number of conceptual clustering methods along dimensions suggested by each of these views. The point the authors ~~we~~ most wish to clarify is that conceptual clustering processes can be explicated as being composed of three distinct but inter-dependent subprocesses: the process of deriving a hierarchical classification scheme; the process of aggregating objects into individual classes; and the process of assigning conceptual descriptions to object classes. Each subprocess may be characterized along a number of dimensions related to search, thus facilitating a better understanding of the conceptual clustering process as a whole.

TABLE OF CONTENTS

1.0 Introduction

Classification is a process critical to the success of an intelligent organism. The ability to classify objects (events, states, observations, etc.) as members of object families or concepts, is the basis of all inferential capacity. Work in Artificial Intelligence has concentrated significantly on developing mechanisms for classification, and the conceptual representations necessary to support these mechanisms. Machine Learning research, specifically work in *learning from examples*, has facilitated a better understanding of processes of *concept identification*, that is the derivation of concepts for a *teacher* imposed classification. Learning from examples however, has not addressed the problem of how a learner can originate classes, but only how conceptual descriptions can be assigned to externally provided classes. Recently methods of *conceptual clustering* have been forwarded, which do provide (partial) solutions to the object class origin problem.

Methods of conceptual clustering are best explicated and compared with respect to two alternative, but complementary views.

Two Views of Conceptual Clustering

1) Methods of conceptual clustering are viewed as extensions or analogs to techniques of *numerical taxonomy*, a collection of methods developed by natural and social scientists used to form classification schemes over data sets.

2) Already alluded to is that conceptual clustering is a form of *concept formation* or *learning by observation* as opposed to learning from examples.

Each of these views has utility in explicating processes of conceptual clustering, and each view will contribute to a unified set of dimensions along which we may characterize various conceptual clustering techniques.

2.0 Conceptual Clustering and Numerical Taxonomy

Conceptual clustering is a process abstraction originally motivated and defined by Michalski (1980) and Michalski and Stepp (1983a) as an extension of processes of numerical taxonomy. Any clustering method, whether it be of the conceptual clustering or numerical taxonomy variety may be abstracted as follows.

The Abstract Clustering Task

Given: A set of symbolically described objects, O.

Task: Distinguish *clusters* (ie. subsets of O),

$C_1, ..., C_n$, such that the set of clusters (ie. a *clustering*) is of high quality (perhaps not optimal) with respect to a clustering quality function.

Methods of numerical taxonomy cluster objects that are symbolically described as sets of *variable-value* pairs (ie. attribute-feature pairs). In methods of numerical taxonomy, the quality of a clustering is a function only of the clusters of the clustering. That is, numerical taxonomy techniques attempt to find a clustering which maximizes a (numeric) quality function of the following form.

$$QUALITY(C_1, C_2, ..., C_n) = f(C_1, C_2, ..., C_n)$$

Despite the usefulness of numerical taxonomy techniques, any such method suffers from a major limitation, in that the resultant clusters may not be well characterized in some human-comprehensible conceptual language. This limitation can be of concern to a data analyst (or learning program) who (which) wishes to abstract the underlying conceptual structure of object groups in order to hypothesize about future observations, or to simply compress the data in an intelligent, easily recoverable way. Michalski (1980) defines conceptual clustering as an extension over the techniques of numerical taxonomy, which directly addresses the problem of determining conceptual representations. In methods of conceptual clustering, the quality of a clustering is dependent on the quality of concepts which may be used to characterize clusters of the clustering (eg. the 'simplicity' of concepts) and/or the *map* between concepts and the clusters they cover (eg. the 'fit' or generality of derived concepts). That is, methods of conceptual clustering seek to obtain clusterings which maximize a quality function of the following form.

$$QUALITY(C_1, C_2, ..., C_n) \\ = f(C_1, C_2, ..., C_n, \text{CONCEPTS})$$

where CONCEPTS is a set of concepts which may be used to describe object clusters.[1]

Conceptual clustering algorithms which have been framed as extensions to numerical taxonomy techniques include **CLUSTER/2** by Michalski and Stepp (1983a, 1983b), **DISCON** by Langley and Sage (1984), and the **RUMMAGE** program by Fisher (1984). A number of other algorithms, although not explicitly labeled conceptual clustering techniques, but which nonetheless can be framed as such, include **GLAUBER** by Langley, Zytkow, Simon, and Bradshaw (1985),

[1] This definition of conceptual clustering differs from but is consistent with Michalski's (1980).

MK10 by Wolff (1980), and Lebowitz' IPP (Lebowitz, 1983) and UNIMEM (Lebowitz, 1982) systems. Each of these systems has a rough analog with some methods of numerical taxonomy which we now touch upon.

The literature on numerical taxonomy distinguishes three classes of methods (Everitt, 1980).

> *Optimization* techniques of numerical taxonomy form a 'flat' (ie. unstructured) set of mutually exclusive clusters (ie. a partition over the input object set). Optimization techniques make an explicit search for a globally optimal K-partition of an object set, where K is a user supplied parameter. This search for globally optimal partitions make optimization techniques computationally expensive, thus constraining their use to small data sets and/or small values of K.
>
> *Hierarchical* techniques form classification trees over object sets, where leaves of a tree are individual objects, and internal nodes represent object clusters. A 'flat' clustering of mutually-exclusive clusters may be obtained from the classification tree by severing the tree at some level. Hierarchical techniques are further divided into *divisive* and *agglomerative* techniques, which construct the classification tree top-down and bottom-up, respectively. Hierarchical techniques depend on 'good' clusterings arising from a series of 'local' decisions. In the case of divisive techniques, a node in a partially constructed tree is divided independent of other (non-ancestrial) nodes of the tree. The use of 'local' decision-making in hierarchical methods make them computationally less expensive than optimization techniques with an associated probable reduction in the quality of constructed clusterings.
>
> *Clumping* techniques return clusterings where constituent clusters possibly overlap. The possibility of cluster overlap stems from independently treating some number of clusters as possible hosts for an object which must be incorporated into a clustering.

We can impose a classification on conceptual clustering methods analogous to the one just discussed for methods of numerical taxonomy. The *Partitioning Module* of CLUSTER/2 by Michalski and Stepp can be viewed as a *conceptual optimization technique* which given an object set to be partitioned and a parameter, K, specifing the number of desired clusters (ie. the partition size), attempts to construct an optimal K-partition of the object set. The partitioning module is computationally expensive and is prohibitive for large values of K. The *Hierarchy-building Module* of CLUSTER/2 is a *conceptual hierarchical technique* which builds a classification tree top-down (ie. it is a divisive technique). In dividing each node in the classification tree, the hierarchy-building module calls the partitioning module for small partition sizes (ie. K), and selects the optimal partition from among these possibilities. Other divisive hierarchical techniques of conceptual clustering include DISCON

and RUMMAGE. Both RUMMAGE and DISCON form *monothetic* classification trees in which any set of siblings in the tree are distinguished by their value along a single variable. In contrast, CLUSTER/2 allows arcs to be labelled by a conjunction of values across several variables, and thus CLUSTER/2 forms *polythetic* classifications. DISCON, unlike both RUMMAGE and CLUSTER/2, discovers an optimal classification tree (in terms of the number of nodes in the completed tree), whereas the latter two algorithms seek only to independently optimize the division of each node, in the hopes that the resultant trees will be of 'high quality'. MK10 by Wolff represents an agglomerative hierarchical technique. *Conceptual clumping techniques* include IPP and UNIMEM by Lebowitz and GLAUBER by Langley et.al.. Each of these systems builds classification schemes equivalent to reentrant, acyclic graphs, where each node represents a cluster, and objects may be included in multiple clusters.

The view of conceptual clustering methods as extensions to methods of numerical taxonomy has served as a vehicle for presenting the input-output behavior of a number of algorithms. For a better understanding the processing characteristics and utility of each of these techniques we turn to the view of conceptual clustering as learning by observation.

3.0 Conceptual Clustering as Learning

An alternative view of conceptual clustering relates this task to the well-studied problem of learning from examples. Both the conceptual clustering task and learning from examples are concerned with formulating some description that summarizes a set of data. In learning from examples, a *tutor* specifies which objects should be assigned to which class, and the learner must characterize each class. In conceptual clustering the learner has the two-fold task of creating object classes as well as characterizing these classes. Thus there are two problems which must be addressed by a conceptual clustering algorithm, one of which is shared by processes of learning from examples.

> The *aggregation problem* is the problem of distinguishing subsets of an initial object set, that is the formation of a set of classes, each defined as an extensionally enumerated set of objects. The aggregation problem is addressed by tasks of conceptual clustering and not by processes of learning from examples which assume a set of classes has been supplied by an external source (ie. a tutor).
>
> The *characterization problem* is the problem of determining characterizations (ie. concepts) for an extensionally represented object class, or each of multiple object classes. This problem has been extensively addressed in work on learning from examples where object classes are presented by a tutor, and the learner is responsible for assigning a conceptual description to each class.

In fact, the characterization problem, as defined here, and the problem of learning from examples are the same. Conceptual clustering processes must address the characterization problem since cluster quality, as we have stated, is dependent on conceptual descriptions which may be used to describe clusters.

We do not mean to imply that the aggregation and characterization (ie. learning from examples) problems are independent, simply that they may be usefully modularlized, thus allowing us to make use of the wealth of information regarding learning from examples in analyzing and formulating methods of conceptual clustering.

Given this view, a natural approach to solving the conceptual clustering problem involves first solving the aggregation problem, and then using traditional methods of learning from examples to solve the characterization problem. In fact, present conceptual clustering algorithms can be framed in this way. For instance, GLAUBER forms classes based on the most commononly occuring relation (defined over an object set) and then characterizes these classes with respect to the remaining relations. MK10 employs a very similar technique (in fact, GLAUBER's method is based on MK10). UNIMEM and IPP construct a number of alternative classes each of which is based on the *predictive* features (ie. variable values) shared by all class members, and characterized by a conjunction of all *predictable* features shared by class members.[2]

Both RUMMAGE and DISCON use a list of user-specified attributes to form possible partitions over an object set. RUMMAGE considers a number of partitions, each implied by the values of a distinct attribute and selects that partition (ie. clustering) which possesses the 'best' conceptual descriptions of objects over the remaining attributes. Thus, RUMMAGE solves the aggregation problem by using individual attribute values to imply possible clusters (the values of a single attribute collectively imply a clustering), and then utilizes a learning from examples subroutine to characterize clusters in terms of the remaining attributes. RUMMAGE applies this method recursively to each of the resulting clusters, thus tracing out a single hierarchical classification scheme. Like RUMMAGE, DISCON uses attribute values to imply possible partitions, thus solving the aggregation problem. Unlike RUMMAGE, DISCON does not construct an explicit description of the devised clusters over the remaining attributes, but simply calls itself recursively on each of the possible clusters, thus forming a classification tree over the objects of each cluster with respect to the remaining attributes. Both RUMMAGE and DISCON are to a greater or lesser extent based on Quinlan's ID3 program for learning from examples (Quinlan, 1983) An abstraction of the aggregation processes of both RUMMAGE and DISCON is given in figure 1.

[2] See Lebowitz (1983) for definitions of predictive and predictable features.

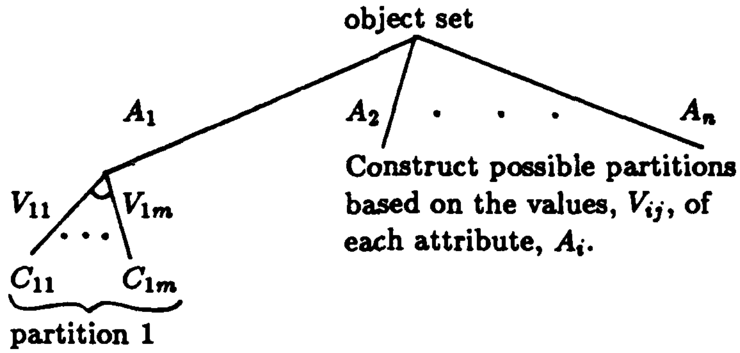


Figure 1 - Aggregation in RUMMAGE and DISCON

The Partitioning Module of Michalski and Stepp's CLUSTER/2 system uses a more experimental solution to the aggregation problem than the systems described above. Given the task of dividing the observed objects into N disjoint classes, the system initially selects N *seed* objects (initially this is done randomly). The system treats each such seed as a positive instance of some class and treats the othe seeds as negative instances of the same class. The program then derives *maximally-general discriminant descriptions* for each class implied by the seeds.[3] The result is that for each seed a number of descriptions (ie. concepts) are derived, each of which covers that seed and no other seed. Each description of each seed also covers some number of non-seed objects which are assigned to the same class as the appropriate seed. Once all objects (seed and non-seed) have been classified with respect to the maximally-general discriminant descriptions, these maximally-general descriptions are 'thrown out', and *maximally-specific characteristic descriptions* are derived for each defined object class. By selecting one description for each seed, a set of (possibly overlapping) clusters, that is a clustering, is implied which classifies the input object set. A pictorial summary of the above process is given in figure 2.

The reasons for this seemingly roundabout means of aggregating and describing object classes are best explicated in Michalski (1980). By first formulating maximally-general descriptions, *any* clustering implied by any combination of maximally-general descriptions (one description for each seed) can be shown to contain *at least one* cluster which covers an arbitrary object. Thus by first formulating maximally-general descriptions, CLUSTER/2 guarentees that every observed object can be classified. Once all objects are classified, derivation of maximally-specific descriptions serve to reduce the possibility of overlapping clusters with respect to unobserved objects. A 'fix-up' operation is then employed to make all possible clusterings mutually-disjoint.

[3] See Michalski (1983) for definitions of discriminant and characteristic descriptions.

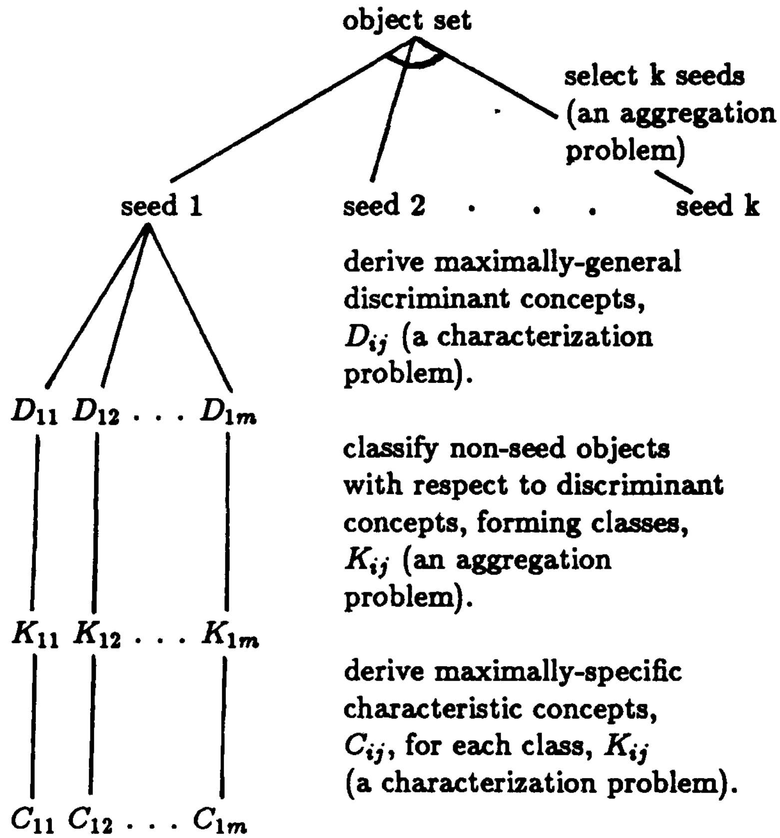


Figure 2 - Aggregation and characterization in
in the Partitioning Module of CLUSTER/2

4.0 Other Dimensions for Characterizing Conceptual Clustering Methods

We have thus far characterized conceptual clustering algorithms in terms of the structuring of the clusterings they produce, and in terms of the ways in which each technique deals with the problems of aggregation and characterization. We now define dimensions relating to search, along which we may describe the subprocesses of conceptual clustering. We begin by discussing dimensions of characterization (ie. learning from examples).

4.1 Searching the Space of Characterizations

As we have seen, the characterization component of the conceptual clustering task is identical to the well-studied task of learning from examples. Thus, we can employ previous results from the machine learning literature in our analysis of this component. For instance, Mitchell (1982), Dietterich and Michalski (1983), and Langley and Carbonell (1984) have proposed various dimensions along which

methods for learning from examples may vary. Mitchell points out that the space of concept descriptions is ordered according to generality. This ordering leads to three alternative schemes for systematically searching the space of hypotheses. First, one may start with a very specific hypothesis, and move toward more general descriptions in search of one that covers the instances; this approach may be called learning by *generalization*. Second, one may start with a very general hypothesis, and move toward more specific descriptions that cover the data; this may be called learning by *discrimination*. Finally, one may search in both directions, hoping to converge on the correct hypothesis; this is Mitchell's *version space* strategy.

Applying this analysis to the characterization components of the existing conceptual clustering systems, we find that UNIMEM/IPP and GLAUBER use generalization in characterizing their groupings. Recall that CLUSTER/2 forms characterizations at two points in its processing: the derivation of maximally-general discriminant concepts uses a discrimination approach; the derivation of maximally-specific characteristic concepts uses a generalization approach. RUMMAGE and DISCON use attribute values to form a number of possible partitions, where each attribute value may be viewed as a maximally-general discriminant concept of the object group it implies. No discrimination or generalization is employed in this process. RUMMAGE does however, use generalization to derive characterizations of object groups over those attributes not used in partitioning the object groups. Wolff's MK10 does not form characterizations *per se*, though it does generate conjunctive descriptions based on co-occurrences.

A second dimension involves the method used to direct search through the space of hypotheses. Some AI systems that learned from examples have used depth-first search to select hypotheses, others have used breadth-first search, while still others have non-exhaustive methods such as beam-search and best-first search. The non-exhaustive methods require some evaluation function to order hypotheses, so the same search technique may give different results depending on the evaluation function it employs. Because of the limited concept languages employed by each of the conceptual clustering systems discussed, there is exactly one maximally-specific concept description for any given object group, which is to say there is no (or only a *degenerate*) search occuring in most cases. Michalski and Stepp's CLUSTER/2 carried out a beam search in deriving maximally-general discriminant concepts, using evaluation functions supplied by the user (such as simplicity of class description). The formation of maximally-specific characteristic descriptions in CLUSTER/2, as with all of the other systems, is deterministic.

Third, one may distinguish between *data-driven* and *model-driven* learning systems. In data-driven systems, the operators for moving through the space of hypotheses require data as input; thus, these data direct the search through the problem space. In model-driven systems, some other knowledge is used to generate new hypotheses, and the data are used only in the evaluation stage. CLUSTER/2,

UNIMEM, GLAUBER, and MK10 employ data-driven characterization methods, while the remaining systems can be viewed as model-driven systems (to the extent that they form characterizations). However, the "models" used by DISCON and RUMMAGE consisted only of a list of attributes that might be used in constructing a classification scheme.

A final dimension concerns whether all observations are processed together, or whether they are handled one at a time. The first situation may be called *non-incremental* learning, and is plausible for modeling scientific data analysis. The vast majority of conceptual clustering systems (CLUSTER/2, DISCON, RUMMAGE, GLAUBER, and MK10) are all non-incremental learning systems. The second situation may be called *incremental* learning, and is more plausible for modeling concept formation based on continuous interaction with one's environment. Of the existing conceptual clustering systems, only UNIMEM and IPP can be viewed as incremental learners. This dimension is associated with the entire conceptual clustering system, not only with the characterization component.

4.2 Searching the Space of Aggregations

As we have seen, conceptual clustering methods solve the aggregation problem as well as the characterization problem, suggesting another set of dimensions along which such methods may differ. In this case, two dimensions present themselves:

- *Search control.* One can imagine a conceptual clustering system systematically considering all possible groupings, evaluating them, and then selecting the best. However, none of the systems we have considered employ such an inefficient approach. Upon inspection, we find that CLUSTER/2 uses a hill-climbing method to home in on an acceptable aggregation, using characterization techniques to evaluate its choices. In contrast, the remaining systems carry out only degenerate searches (of depth one) through the aggregation space, since they select their groupings in a one-step process.
- *Nature of the operators.* In order to understand why RUMMAGE, DISCON, and most other systems require only one-step searches, we must examine the operators they use to generate candidate groupings. RUMMAGE and DISCON both require a user-specified list of attributes and their values; by selecting an attribute, these systems automatically generate a candidate grouping (one for each value of the attribute), which can then be evaluated. GLAUBER, MK10, and UNIMEM/IPP all accomplish the same effect in a more data-driven manner. Only in CLUSTER/2 do we find a less constrained operator, which selects seed objects that may or may not lead to a useful characterization.

4.3 Searching the Space of Hierarchies

We have seen that unlike systems that learn from examples, conceptual clustering methods must also determine their own aggregations. However, there remains another issue that distinguishes conceptual clustering from the task of learning from examples. In the latter, one is generally concerned with forming concepts at a single level, while conceptual clustering usually focuses on generating *hierarchies* of concepts. Some numerical taxonomy methods (the optimization techniques) generate only single level groupings, but most methods arrive at some tree of groupings.

The implication for our analysis of conceptual clustering methods is clear – the search for aggregations and the search for characterizations are embedded within a higher level search through the space of classification trees. Moreover, we can classify the existing clustering systems in terms of two additional dimensions. These are:

- *Direction of the search.* Upon examining the existing conceptual clustering systems, we find that divisive (top-down) methods have been used by the majority, including CLUSTER/2, DISCON, and RUMMAGE. These systems start with a single class of observations, and proceed by subdividing the instances into classes, these classes into subclasses, and so forth. However, one can also imagine methods that begin with separate "classes" for each observation, joining these classes together to form larger classes, and joining these classes in turn. Such bottom-up (agglomerative) methods have been used by a minority of conceptual clustering systems, including GLAUBER and MK10. Other arrangements are also possible; for example, Mervis and Rosch (1981) have suggested an approach where one first forms classes of medium generality, and later forms both more general and more specific classes. UNIMEM/IPP behaves in roughly this manner and at any point in its processing classes of greater or lesser generality than existent classes may be added to the classification.

- *Search control.* Conceptual clustering systems must somehow direct their search through the space of hierarchies. Upon examining the existing systems, we find that CLUSTER/2, RUMMAGE, GLAUBER, and MK10 carry out only degenerate searches through this space. The reason is that their operators consist of techniques for finding optimal aggregations and characterizations. Search is involved at these lower levels, but the result is an optimal extension to the hierarchical tree. In contrast, DISCON has degenerate search schemes at these lower levels, but carries out a best-first search through the space of hierarchies. It accomplishes this through an exhaustive look-ahead process, evaluating entire sub-trees and preferring those containing fewer nodes. UNIMEM and IPP also carried out search at this level,

entertaining multiple organizations (thus using a form of beam search); however, these organizations might be revised later in the search, so backup was allowed.

Although these dimensions are similar to those presented for the characterization problem, it is important to note that the current dimensions are separate from those for characterization. For instance, CLUSTER/2 employed beam search to find maximally-general discriminant descriptions, but employed only a degenerate search for determining the best hierarchy.

5.0 Concluding Remarks

We have discussed the mechanics of a number of conceptual clustering methods and defined dimensions which serve to clarify the differences and similarities between methods. Our bias has been that further work in conceptual clustering is best facilitated by first understanding these processes in terms of well-understood concepts. Following Michalski (1980), we have presented conceptual clustering as an extension of numerical taxonomy. Further, by framing conceptual clustering as a composition of aggregation and characterization processes, we have shown a relationship between conceptual clustering and methods of learning from examples. This dichotomy has led to a view of conceptual clustering processes as conducting a three-tiered search: a search through a space of hierarchies; a search through a space of possible aggregations; and a search through a space of conceptual descriptions.

It is our view that explicating conceptual clustering as multi-layered search will not only ease comprehension of existing methods, but facilitate work in a number of still open problem areas.[4] One problem concerns the task of clustering *structured* objects, where object descriptions allow relations to be represented between attribute values of an object. Vere's THOTH system (Vere, 1978) is currently being investigated as a basis for a conceptual clustering system for structured objects. THOTH discovers a minimal set of generalizations which cover a given set of *relational production* instances, where each production instance is a (before) state - (after) state pair. Each state representation is equivalent to a structured object representation. THOTH traces out a hierarchical classification bottom-up and in many ways resembles an agglomerative approach to conceptual clustering. A second area of interest to us concerns the problem of utilizing information on the *functionality* of objects to aid the formation of useful clusters. An approach suggested in discussion by Nelson (1977) involves using domain-specific knowledge of object functionality to guide the search for possible aggregates, and to use perceptual information as the basis of characterization. Distinct forms of knowledge may serve to guide the search

[4] See (Langley and Carbonell, 1984; Michalski and Stepp, 1983b) for comprehensive discussions of open problems in conceptual clustering.

for hierarchies. By distinguishing levels of search we can more easily motivate and express the rules, heuristics, and descriptive languages utilized at different levels.

REFERENCES

[1] Dietterich, T. and Michalski, R. "A Comparative Review of Selected Methods of Learning from Examples" In *Machine Learning*, Michalski, R., Carbonell, J., and Mitchell, T. (eds.), Tioga Publishing Co., Palo Alto, CA., 1983.

[2] Everitt, B. *Cluster Analysis*, Heinemann Educational Books, Ltd., 1980.

[3] Fisher, D. "A Hierarchical Conceptual Clustering Algorithm", Technical Report, Department of Information and Computer Science, University of California, Irvine, 1984.

[4] Langley, P. and Carbonell, J. "Approaches to Machine Learning" In *Journal of the American Society for Information Science 35, 5* (1984), 306-316.

[5] Langley, P. and Sage, S. "Conceptual Clustering as Discrimination Learning" In *Proceedings of the Fifth Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, (1984).

[6] Langley, P., Zytkow, J., Simon, H., and Bradshaw, G. "The Search for Regularity: Four Aspects of Scientific Discovery" In *Machine Learning, Volume 2* (to appear), Michalski, R., Carbonell, J., and Mitchell, T. (eds.), Tioga Publishing Co., Palo Alto, CA., 1985.

[7] Lebowitz, M. "Correcting Erroneous Generalizations" In *Cognition and Brain* Theory 5, 4 (1982), 367-381.

[8] Lebowitz, M. "Generalization from Natural Language Text" In *Cognitive Science 7, 1* (1983), 1-40.

[9] Mervis, C. and Rosch, E. "Categorization of Natural Objects" In *Annual Review of Psychology*, 1981, pp. 89-115.

[10] Michalski, R. "Knowledge Acquisition Through Conceptual Clustering: A Theoretical Framework and Algorithm for Partitioning Data into Conjunctive Concepts" In *International Journal of Policy Analysis and Information Systems 4, 3* (1980), 219-243.

[11] Michalski, R. "A Theory and Methodology of Inductive Learning" In *Machine Learning*, Michalski, R., Carbonell, J., and Mitchell, T. (eds.), Tioga Publishing Co., Palo Alto, CA., 1983.

[12] Michalski, R. and Stepp, R. "Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy" In *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5,4(1983a), 396-409.

[13] Michalski, R. and Stepp, R. "Learning from Observation: Conceptual Clustering" In *Machine Learning*, Michalski, R., Carbonell, J., and Mitchell, T. (eds.), Tioga Publishing Co., Palo Alto, CA, 1983b.

[14] Mitchell, T. "Generalization as Search" In *Artificial Intelligence 18, 2* (1982), 203-226.

[15] Nelson, K. "Some Evidence for the Cognitive Primacy of Categorization and its Functional Basis" In *Thinking: Readings in Cognitive Science*, Johnson-Laird, P. and Wason, P. (eds.), Cambridge University Press, New York, 1977.

[16] Quinlan, J. "Learning Efficient Classification Procedures and Their Application to Chess End Games" In *Machine Learning*, Michalski, R., Carbonell, J., and Mitchell, T. (eds.), Tioga Publishing Co., Palo Alto, CA., 1983.

[17] Vere, S., "Inductive Learning of Relational Productions" In *Pattern-Directed Inference Systems*, Waterman, D. and Hayes-Roth, F. (eds.), Academic Press, Inc., Orlando, Florida, 1978.

[18] Wolff, J. "Data Compression, Generalisation, and Overgeneralisation in an Evolving Theory of Language Development" In *Proceedings of the AISB-80 Conference on Artificial Intelligence* (1980), W1-10.

END

FILMED

10-85

DTIC